

IN THE CLAIMS

5 Please amend Claims 1-5, 7-9, 11-20, 22-31, 33-38, 40-42, 44-46, and 48-49, and add new Claim 50, as follows:

1. (Currently amended) An embedded-DRAM (dynamic random access memory) processor comprising:

10 a DRAM array comprising a plurality of random access memory cells arranged in rows and columns;

a ~~set of~~ row address ~~register registers~~ that holds a pointer that points to a row of the DRAM array;

15 one or more sets of registers, each of said sets of registers capable of being loaded or stored in response to a single latch signal; and

an instruction set which includes:

(i) at least one command to perform arithmetic on said row address registers;

20 (ii) a command to precharge (activate) the row ~~rows~~ pointed to by said row address ~~registers~~ register;

(iii) a command to deactivate said row pointed to by said row address ~~registers~~ register after it had been precharged by the command to precharge;

25 (iv) a command to load a plurality of words of a the precharged row ~~designated by said row address registers~~ into designated sets of data registers; and

(v) a command to load selected columns of the precharged row ~~rows~~ pointed to by said row address registers into designated sets of data registers, said selection based on bits in a mask.

30 2. (Currently amended) The embedded-DRAM processor according to Claim 1, further comprising:

first and second sets of functional units, said first and second sets of functional units having respective first and second instruction sets and capable of accessing ~~said~~ first and second ~~register~~ sets of said registers;

5 a command to select one of said first and second sets of registers to be an architectural set of registers accessible to said first set of functional units;

a command to deselect the other of said first and second sets of registers so that it is no longer an architectural register set accessible to said first set of functional units;

a command to select one of said first and second sets of registers to be an architectural set of registers accessible to said second set of functional units; and

10 a command to deselect the other one of said first and second sets of registers so that it is no longer an architectural register set accessible to said second set of functional units.

3. (Currently amended) The embedded-DRAM processor according to Claim 1, further comprising:

15 first and second sets of functional units, said first and second sets of functional units having respective first and second instruction sets and accessing ~~said~~ first and second ~~register~~ sets of registers; and

20 a command which selects one of said first and second sets of registers to be an architectural set of registers accessible to said first set of functional units, and, at the same time, which deselects the other one of said one of said first and second sets of registers to be an architectural set of registers accessible to said second set of functional units.

4. (Currently amended) The embedded-DRAM processor according to Claim 1, further comprising:

25 first and second sets of functional units, said first and second sets of functional units having respective first and second instruction sets and accessing ~~said~~ first and second ones of said register sets; and

whereby said first and second instruction sets are subsets of said instruction set of said embedded-DRAM processor.

5. (Currently amended) The embedded-DRAM processor according to Claim 4, whereby said second set of functional units comprises a functional unit that is a multi-issue functional unit, and further comprises:

a dispatch unit;

5 a plurality of functional units which each execute a respective instruction stream as dispatched by said dispatch unit.

6. (Original) The embedded-DRAM processor according to Claim 1, further comprising a plurality of DRAM arrays.

10 7. (Currently amended) The embedded-DRAM processor according to Claim 1, further comprising:

at least one functional unit;

15 whereby said one or more sets of registers comprise a plurality of register files, each of said register ~~file~~ files comprising a parallel access port operative to load or store contents of said register file in a single cycle from or to a DRAM row as selected by said row-address register, each said register file further comprising at least a second access port operative to transfer data between said functional unit and a selected subset register in said register file.

8. (Currently amended) The embedded-DRAM processor according to Claim 7, further comprising:

20 a second functional unit;

25 whereby said first functional unit executes a first command to perform logical processing on the contents of one or more registers within a selected active one of said register sets, and said second functional unit [[a]] executes a second command to parallelly transfer data between a selected inactive one of said register sets and said DRAM array.

9. (Currently amended) The embedded-DRAM processor according to Claim 8, ~~whereby~~ wherein said first and second functional units execute said first and second commands substantially contemporaneously.

10. (Original) The embedded-DRAM processor according to Claim 8, further comprising:

a first software module comprising a set of data manipulation commands, said first software module executed by said first functional unit; and

5 a second software module comprising a set of parallel data transfer commands, said second software module being executed by said second functional unit;

whereby said second software module operates in support of said first software module to prefetch data from said DRAM array into one of said register files in advance of said data being needed by said first software module.

10 11. (Currently amended) The embedded-DRAM processor according to Claim 10, wherein:

said first software module contains an instruction that reference registers within an architectural register set visible to said first functional unit, whereby said architectural register set corresponds to at least partially to said one of said register files that is in an active state;

15 said second software module contains instructions that cause data to be transferred between an inactive register set and said DRAM array, and second software module also executes a command to toggle a selected register set between said active and inactive states.

20 12. (Currently amended) The embedded-DRAM processor according to Claim 1, further comprising:

first and second sets of functional units, said first and second sets of functional units having respective first and second instruction subsets;

25 whereby said first instruction subset includes said command (i) and the second instruction subset includes said commands (ii), (iii), (iv) and (v) ~~(iv)~~.

13. (Currently amended) An embedded-DRAM (dynamic random access memory) processor comprising:

a DRAM array comprising a plurality of random access memory cells arranged in rows and columns;

30 a row address register that holds a pointer that points to a row of the DRAM array;

one or more sets of data registers, each of said sets of data registers capable of being loaded or stored in response to a single latch signal;

a bit mask to select one or more data locations within at least one of said register sets; and

5 an instruction set which comprises at least:

- (i) a command to perform arithmetic on said row address register;
- (ii) a command to precharge (activate) a the row pointed to by said row address register
- (iii) ~~a command to deactivate a row pointed to by said row address~~
10 ~~register;~~
- (iv) a command to load a set of selected elements of the precharged row pointed to by said row address register into a selected set of said data registers, said selection based on bits in said bit mask[.];

15 wherein the command to precharge is executed to precharge the row prior to the command to load so that at the time the command to load is issued, the command to load can execute without the need to wait for the row to precharge.

14. (Currently amended) The embedded-DRAM processor according to Claim 13, ~~whereby~~ wherein said load command causes an entire row that was previously precharged to be loaded.

20 15. (Currently amended) The embedded-DRAM processor according to Claim 13, ~~whereby said load command causes a subset of a row that was previously precharged to be loaded.~~ further comprising:

25 (v) a command to store a selected set of said data registers to a set of selected elements of the precharged row pointed to by said row address register into a selected set of said data registers, said selection based on bits in said bit mask;

wherein after the command to precharge the row has been executed, the command to load and the command to store can each be executed a plurality of times without needing to recharge the precharged row.

30 16. (Currently amended) An embedded-DRAM (dynamic random access memory) processor comprising:

a DRAM array comprising a plurality of random access memory cells arranged in rows and columns;

a row address register that holds a pointer that points to a row of the DRAM array;

first and second ~~registers~~ register files, each of said register files having a plurality
5 of data registers capable of being loaded or stored in response to a single latch signal,
each of said register files also being capable of being placed into an active state and an
inactive state;

a set of functional units that perform logical operations on data accessed from a
set of architectural registers, wherein registers placed into the active state appear as
10 architectural registers to a set of functional units, and registers in the inactive state the
registers are not accessible by the functional units;

a bit mask to select one or more locations within at least one of said register files;
and

an instruction set which comprises at least:

- 15 (i) a command to perform arithmetic on said row address register; and
(ii) a command to load a set of selected elements of the row pointed to
by said row address register into a selected set of said data
registers, said selection based on bits in said bit mask, and the
selected set of said data registers being in the inactive state.

20 17. (Currently amended) The embedded-DRAM processor of Claim 16, ~~whereby~~
wherein the instruction set further comprises:

- (iii) a command to toggle a register set between said active and inactive
states.

18. (Currently amended) The embedded-DRAM processor of Claim 17, ~~whereby~~
25 wherein said toggle command causes said first register file to toggle from the inactive state to the
active state and also causes the second register file to toggle from the active state to the inactive
state.

19. (Currently amended) The embedded-DRAM processor of Claim 16, ~~whereby~~
wherein the instruction set further comprises:

- 30 (iii) a command to manipulate the bits in the bit mask.

Application No. : 10/074,779
Filed : February 13, 2002

20. (Currently amended) The embedded-DRAM processor according to Claim 16, further comprising:

first and second sets of functional units; ~~said first and second sets of functional units having respective~~

5 first and second ~~instruction sets and~~ sets of instructions capable of accessing said first and second register sets; and

said instruction set further comprises at least:

(iii) a command to select one of said first and second sets of registers to be an architectural set of registers accessible to said first set of
10 functional units; and

(iv) a command to select one of said first and second sets of registers to be an architectural set of registers accessible to said second set of functional units.

21. (Original) The embedded-DRAM processor according to Claim 20, said instruction
15 set further comprising:

(v) a command to deselect the other of said first and second sets of registers so that it is no longer an architectural register set accessible to said first set of functional units; and

(vi) a command to deselect the other one of said first and second sets of
20 registers so that it is no longer an architectural register set accessible to said second set of functional units.

22. (Currently amended) The embedded-DRAM processor according to Claim 20, ~~whereby~~ wherein at least one of said sets of functional units contains a single functional unit.

23. (Currently amended) An embedded-DRAM (dynamic random access memory)
25 processor comprising:

a DRAM array comprising a plurality of random access memory cells arranged in rows and columns;

a row address register that holds a pointer that points to a row of the DRAM array;

first and second registers files, each of said register files capable of being loaded
30 or stored in response to a single latch signal, ~~each of said register files also being capable of being placed into an active state and an inactive state;~~ and

first and second of functional units, said first and second functional units having respective first and second instruction sets and capable of accessing said first and second register sets files;

5 ~~whereby~~ wherein said first and second ~~registers~~ register files comprise a parallel access port operative to parallelly transfer contents of one of said register ~~file~~ files between a DRAM row as selected by said row-address register, ~~each~~ said first and second register ~~file~~ files further comprising at least a second access port operative to transfer data between a selected one of said register ~~file~~ files and said second functional unit;

~~whereby~~ wherein said first instruction set comprises at least:

- 10 (i) a command to manipulate data in a data register within a register file; and

~~whereby~~ wherein said second instruction set comprises at least:

- (ii) a command to perform arithmetic on said row address register;
15 (iii) a command to load the entire row pointed to by said row address register into a selected set of registers of said register files.

24. (Currently amended) The embedded-DRAM processor according to Claim 23, ~~whereby~~ wherein said first and second functional units each respectively execute a command from said first and second instruction sets substantially contemporaneously.

25. (Currently amended) The embedded-DRAM processor according to Claim 24,
20 further comprising:

a first software module comprising [[a]] data manipulation commands drawn from said first instruction set, said first software module executed by said first functional unit; and

25 a second software module comprising a parallel data transfer command drawn from said second instruction set, said second software module being executed by said second functional unit;

whereby said second software module operates in support of said first software module to prefetch data from said DRAM array into one of said register files in advance of said data being needed by said first software module.

Application No. : 10/074,779
Filed : February 13, 2002

26. (Currently amended) The embedded-DRAM processor of Claim 23, ~~whereby~~
wherein the second instruction set further comprises:

(iv) a command to toggle a register set between said active and inactive states.

27. (Currently amended) The embedded-DRAM processor of Claim 26, ~~whereby~~
5 wherein said toggle command causes said first register file to toggle from the inactive state to the
active state and also causes the second register file to toggle from the active state to the inactive
state.

28. (Currently amended) An embedded-DRAM (dynamic random access memory)
processor comprising:

10 a DRAM array comprising a plurality of random access memory cells arranged in
rows and columns;

first and second dual-port registers files, each of said register files capable of
parallelly transferring data between a row of said DRAM array, each of said register files
also being capable of being placed into an active state and an inactive state; and

15 first and second of functional units, said first and second functional units having
respective first and second instruction sets that operate on registers of an architectural
register set, said architectural register set comprising one of the first and second dual-port
registers files that is currently in the active state;

~~whereby~~ wherein said first instruction set comprises at least:

20 (i) a command to manipulate data in a data register within a register
file; and

~~whereby~~ wherein said second instruction set comprises at least:

(ii) a command to unidirectionally transfer data between a row of said
DRAM array and a selected inactive data register file;

25 (iii) a command to place said inactive register file into said active state,
~~whereby~~ wherein when the register set is activated, it becomes an
architectural register set of said first functional unit.

Application No. : 10/074,779
Filed : February 13, 2002

29. (Currently amended) The embedded-DRAM processor of Claim 28, ~~whereby~~
wherein said command to unidirectionally transfer data causes data to be transferred from a row
of the DRAM array to said selected inactive data register file.

30. (Currently amended) The embedded-DRAM processor of Claim 28, ~~whereby~~
wherein said command to unidirectionally transfer data causes data to be transferred from said
selected inactive data register file to a row of the DRAM array.

31. (Currently amended) The embedded-DRAM processor of Claim 28, ~~whereby the~~
wherein said command to place the selected inactive register file into the active state is a
command that also causes the remaining register file to toggle from the active state into the
inactive state.

32. (Original) The embedded-DRAM processor of Claim 28, further comprising:
at least one additional register file;

whereby the said command to place the selected inactive register file into the active state
is a command that also causes a selected other register file to toggle from the active state into the
inactive state.

33. (Currently amended) The embedded-DRAM processor of Claim 28, further
comprising:

at least one [[a]] row address pointer, whereby at least one command in said
second instruction set uses said row address pointer to ~~identify~~ point to said selected
register file; and

the second instruction set further comprises:

(iv) a command to manipulate the at least one row address pointer.

34. (Currently amended) The embedded-DRAM processor of Claim 28, further
comprising:

at least one [[a]] bit mask; and

the second instruction set further comprises:

(iv) a command to move a subset of elements between a selected register file and a selected row of said DRAM array, whereby said subset is identified by said bit mask.

35. (Currently amended) The embedded-DRAM processor according to Claim 28,
5 ~~whereby~~ wherein said first functional unit is a multi-issue functional unit and further comprises:
a dispatch unit;
a plurality of functional units that each execute a respective instruction stream as dispatched by said dispatch unit.

36. (Currently amended) The embedded-DRAM processor according to Claim 28,
10 further comprising:
a first software module comprising a set of data manipulation commands drawn from said first instruction set, said first software module executed by said first functional unit; and

15 a second software module comprising a set of parallel data transfer commands drawn from said second instruction set, said second software module being executed by said second functional unit;

~~whereby~~ wherein said second software module operates in support of said first software module to prefetch data from said DRAM array into one of said register files in advance of said data being needed by said first software module.

20 37. (Currently amended) The embedded-DRAM processor according to Claim 28, wherein:

said first software module contains an instruction that ~~reference~~ references registers within an architectural register set visible to said first functional unit, whereby said architectural register set corresponds to at least partially to said one of said register files that is in an active state;

25 said second software module contains instructions that cause data to be transferred between an inactive register set and said DRAM array, and second software module also executes a command to toggle a selected register set between said active and inactive states.

Application No. : 10/074,779
Filed : February 13, 2002

38. (Currently amended) The embedded-DRAM processor according to Claim 28, whereby each of said register files contain a number of words, N, matched to the number of words in of a row of said DRAM array, and said ~~parallel load and store operations involve~~ unidirectional transfer comprises moving said selected row in its entirety to said selected register
5 file.

39. (Original) The embedded-DRAM processor according to Claim 28, further comprising:

a mask and switch unit interposed between said DRAM array and at least one of said register files.

10 40. (Currently amended) The embedded-DRAM processor according to Claim 28, ~~whereby~~ wherein said second set of instructions comprises:

a command to cause data to be moved from one register to another within a given one of said register files.

15 41. (Currently amended) The embedded-DRAM processor according to Claim 28, ~~whereby~~ wherein said second instruction set is used to implement an intelligent caching scheme, whereby said register files act as a cache and said second set of instructions are executed in lieu of a standard cache that maintains most recently used data and enforces a set associative or a direct-mapped caching policy.

20 42. (Currently amended) The embedded-DRAM processor according to Claim 28, further comprising:

an instruction register coupled to receive instructions from said instruction set, said instruction register operative to hold an instruction to be executed by ~~said~~ a data assembly unit; and

a local program memory coupled to said instruction register;

25 whereby said second functional unit corresponds to ~~[[a]]~~ said data assembly unit, and said data assembly unit receives an instruction from said second instruction set that causes a separate control thread of instructions to be accessed from said local program memory and executed by said data assembly unit.

43. (Original) The embedded-DRAM processor of claim 42, further comprising:

a prefetch unit that prefetches instructions from the first and second instruction sets from a single very long instruction word (VLIW) instruction memory; and
a dispatch unit that dispatches instructions from the first instruction set to the functional units and dispatches instructions from the second instruction stream to the data assembly unit.

44. (Currently amended) The embedded-DRAM processor according to Claim 28, ~~whereby~~ wherein said second functional unit monitors execution activity of instructions in said first instruction set and said second instruction set further comprises:

(iv) a command to precharge a row of the DRAM array;

whereby the second functional unit executes a speculative precharging to prevent program delays due to DRAM row precharging.

45. (Currently amended) An embedded-DRAM (dynamic random access memory) processor comprising:

a DRAM array comprising a plurality of random access memory cells;

first and second dual-port registers files, whereby the first port of each of said register files is a parallel access port and is parallelly coupled to said DRAM array, each of said register ~~file~~ is files being capable of being placed into an active state and an inactive state;

~~at least one~~ a functional unit that executes a first program, said functional unit coupled to said second port of said register files, said functional unit responsive to commands exclusively involving architectural register operands that map onto to the registers within a register file that is in the active state;

a data assembly responsive to an instruction set comprising at least:

- (i) a command that causes data to be moved between the DRAM array and a register file that is in the inactive state; and
- (ii) a command that causes said register file in the inactive state to assume the active state and said register file in the active state to assume the inactive state.

46. (Currently amended) In a digital processor comprising a DRAM array having a plurality of random access memory cells arranged in rows and columns, ~~a set of at least one row address registers~~ register, one or more sets of registers each the at least one row address register

Application No. : 10/074,779
Filed : February 13, 2002

capable of being parallelly loaded or stored ~~in response to a latch signal~~, a method of ~~processing data~~ comprising:

performing arithmetic on said at least one row address register[[s]] in order to manipulate a pointer that points to a row of the DRAM array;

5 speculatively precharging (activating) a row ~~rows~~ pointed to by said row address register[[s]] based upon a possible anticipated need to perform one or more load and or store operations that would access said row; and

in response to a separate command executed after the speculatively precharging, loading a plurality of words of a row designated by said row address registers into
10 designated sets of data registers.

47. (Original) The method of Claim 46, further comprising deactivating rows pointed to by said row address registers.

48. (Currently amended) In a digital processor comprising a DRAM array having a plurality of random access memory cells arranged in rows and columns, a set of row address
15 registers, one or more sets of data registers each capable of being loaded or stored in response to a latch signal, a method of processing data comprising:

 performing arithmetic on said row address registers;

 precharging (activating) rows pointed to by said row address registers; and

in response to an instruction issued after the precharging, loading selected
20 columns of rows pointed to by said row address registers into designated sets of said data registers, said selection based on bits in a mask.

49. (Currently amended) In a digital processor comprising a DRAM array having a plurality of random access memory cells arranged in rows and columns, first and second dual-
port registers files each capable of (i) parallel transfer of data between a row of said DRAM
25 array, and (ii) being placed into an active state and an inactive state, and first and second functional units, a method for processing data comprising:

 manipulating data in a data register within a register file that is in an the active state using said first functional unit; and

 using said second functional unit;

30 (a) unidirectionally transferring data between a row of said DRAM array and a selected inactive data register file; and

(b) placing said inactive register file into said active state, whereby when the register file is activated, it becomes an architectural register set of said first functional unit.

50. (New) An embedded-DRAM (dynamic random access memory) processor

5 comprising:

a DRAM array comprising a plurality of random access memory cells arranged in rows and columns;

a row address register that holds a pointer that points to a row of the DRAM array;

10 first and second registers files, each of said register files capable of being loaded or stored in response to a single latch signal, each of said register files also being capable of being placed into an active state and an inactive state; and

first and second of functional units, said first and second functional units having respective first and second instruction sets and capable of accessing said first and second register sets while in said active state;

15 wherein said first and second registers files comprise a parallel access port operative to parallelly transfer contents of said register file between a DRAM row as selected by said row-address register, each said register file further comprising at least a second access port operative to transfer data between a selected register file and said second functional unit;

20 wherein said first instruction set comprises at least:

a command to manipulate data in a data register within a register file; and

wherein said second instruction set comprises at least:

a command to perform arithmetic on said row address register; and

25 a command to load the entire row pointed to by said row address register into a selected set of registers of said register files that is currently in the inactive state.